

REMARKS/ARGUMENTS

This Amendment is in response to the Final Office Action mailed February 3, 2005. Claims 1-13 were examined in the Office Action and all have been rejected. Claim 11 is amended herein. There are no new claims added. Reexamination and reconsideration are respectfully requested.

Claim Amendments

Claim 11 is amended herein to more particularly point out the functionality of the data structure stored on the computer readable medium.

Claim Rejections – 35 U.S.C. § 101

Claims 11-13 stand rejected under 35 U.S.C. § 101 because they represent mere arrangement of data, do not define a functional interrelationship among that data and the computing process performed when reutilizing that data and as such do not implement a statutory process.

Applicants herein amend claim 11 to particularly point the functional interrelationship between the data and the computer system. Therefore, under *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium and that increases computer efficiency held statutory subject matter) and *in re Warderdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to data structure stored in memory held statutory product-by-process claim). Applicants believe amended claim 11 is now in a condition for allowance. See, Manual for Patent Examining Procedure, Section 2106(a).

Therefore Applicants respectfully request the Examiner to withdraw this rejection and find claims 11-13 in a condition for allowance.

Claim Rejections under 35 U.S.C. § 102(e) of Claims 1-2 and 7-8

Claims 1-2 and 7-8 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over Lee (USPN 6,732,362). In the rejection, the Examiner cites Lee's discussion of object-oriented programming and object classes from which objects are read into memory (i.e. instantiated) as

anticipating schema documents that define property sheets and property pages. The Applicants respectfully traverse the Examiner's rejection.

The objects cited by the Examiner in Lee conform to the Component Object Model (COM) of object-oriented programming. As well described in Lee, objects are discrete data structures that expose both functions and properties (data) through an "interface." An object is "instantiated" when a class is read into memory and values are provided for the properties. The section cited most often by the Examiner, col. 3, lines 20-65, describe the common act of instantiating an object in which it is verified that all necessary values for the object are provided with the command to instantiate the object.

The schema documents claimed in claims 1-2 and 7-8 are not objects and differ from objects in several important respects. First, Schema documents do not expose functions and are not "instantiated". The schema documents in the claims 1-2 and 7-8 are an abstraction of data concerning the objects managed by resources of the network. The first schema document does not correspond to an object class, in that it is not "instantiated" and cannot be called on to create an object instance.

Second, the property sheet as claimed is modified by each second schema document received to include an additional property page. Object classes in COM and as described in Lee, are not modified but are typically stored as read only data that is read only when instantiating an object instance. Furthermore, other than changing values contained in an object's property, instantiated objects cannot be modified after being instantiated.

Third, the property sheet as claimed is modified by second schema documents provided by different resources, such as applications or hardware resources on the network. It should be clear to the Examiner that it is neither possible nor desirable for an object class provided by a first resource (such as an application) to be modified by a second resource as claimed. Consider, for example, what would happen if an application, such as a wordprocessor, object class was changed by each additional resource added to the computing network. The result would be classes that, when instantiated, would no longer create object instances usable by the original application because of the modifications.

Fourth, the Examiner makes no attempt to identify examples in the prior art of property pages supplied by a hardware resource. Rather, the Examiner points to an installation message that contains data being passed to a database management system. The examiner makes no

attempt to show that the data passed conforms to any form predetermined by a first schema as defined by a first resource. While data from the installation message may end up in an object instantiated from the database's object class library, in Lee the structure of the message is dictated by the operating system or the resource. Furthermore, the property pages as claimed are more than just data, they are displayable pages (conforming to a schema) as described in claim 3 that are stored with the property sheet. Therefore, Lee's installation message does not anticipate receiving a property page from hardware resource as claimed.

The Applicants note that the Examiner also cited Calder (USPN 5,949,417) as relating to property sheets and property pages. In Calder, the property pages are used as separate user interfaces in a graphical user interface system that may be modified by a user. However, the Examiner rightly did not find the above claim elements in Calder. Calder does not show first schema documents and second schema documents as claimed. Nor does Calder describe receiving schema documents from different resources and modifying a property sheet provided by a first resource to create a property sheet that represents an object as claimed.

The Examiner is directed to the discussion in Calder, col. 7, lines 19-61, in which object-oriented programming is discussed in further detail, specifically outlining the difference in the object-oriented programming model and how it could be applied to creating user interfaces for an application using property sheets and property pages. Calder provides further detail on the differences between object classes and property sheets.

For the above reasons, Applicants believe that neither Lee nor Calder show all the limitations and, therefore, independent claim 1 and its dependent claims 2-10 are in an allowable form. Therefore, Applicants respectfully request the Examiner withdraw the rejection and find claims 1-10 in a condition for allowance.

Claim Rejections of claims 3-6, 9 and 10 under 35 U.S.C. § 103(a)

The Examiner further rejected claims 3-5, and 9 under 35 U.S.C. 103(a) over Lee in view of Calder. The Examiner further rejected claims 6 and 10 over Lee in view of the W3C reference.

However, for the reasons pointed out above, neither Lee nor Calder - alone or in combination - anticipate all the elements of the independent claims. Furthermore, the W3C reference does not teach or disclose any elements beyond that of Calder and Lee other than the

use of XML. The W3C certainly does not disclose first and second schema documents as claimed, nor does it disclose modifying a property sheet with property pages provided by different resources to create a property sheet that represent an object managed by a first resource.

Claim Rejections of claims 11-13 under 35 U.S.C. § 103(a)

The Examiner further rejected claims 11-13 under 35 U.S.C. 103(a) over Lee in view of Calder. Applicants traverse this rejection.

Calder discloses property sheets and property pages as part of a GUI. In Calder, the property sheets and pages are all application-specific, i.e., the property sheet and its constituent property pages are created by one resource. Calder's invention is the ability to separate – “tear away” a property sheet from the currently displayed location, which allows a user to view two property pages simultaneously (whereas normally one page would obscure another). In Calder's vernacular, a second “property sheet system” is created at the new location so that two systems are simultaneously displayed to the user. Calder's property sheet and property pages are the GUI for an application.

Applicants first refer the Examiner to the above arguments wherein it is explained the object classes and objects of Lee are not the same as property sheets and property pages.

Applicants further point out that neither Calder nor Lee disclose property pages provided by different resources, as in both Calder and Lee property sheets or objects are provided by one particular resource only for use by that resource.

Finally, Calder and Lee also do not teach or disclose property sheets that “represent an object maintained by one of the multiple resources”. Both Calder and Lee describe systems that are application specific. In Calder, the property sheet and its pages are provided by, and only associated with, one application. In Lee, object classes are provided by and for use with the database application.

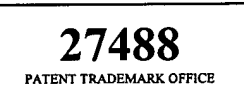
Therefore, as neither Calder nor Lee alone or in combination describe all of the elements of the claimed invention, Applicants respectfully request that the Examiner withdraw this rejection of claims 11-13 and find said claims in a condition for allowance.

Conclusion

Claims 1-13 currently remain pending in the present application. In light of the above remarks, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve those issues.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

Dated: _____



Respectfully submitted, _____

George C. Lewis, #53,214
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1639